Amendments to the Specification:

Please replace the first paragraph on page 5, with the following amended paragraph:

-- Management consoles 125, 130 provide connections to the local, and optionally to the remote disk system, using LAN 133, proprietary connection 135, SCSI, Fibre or ESCON, or other well known technique. An administrator manages the disk systems through this management consoles 125, 130. If desired, the management console 125 for the local disk system also may connect to the remote disk system. The connection between the local and remote disk systems may comprise ESCON, SCSI, LAN/WAN or Fibre 140, or combination of them, for example, using a gateway appliance. As shown in ~~Figure1~~ Figure 1, a key is assigned to a volume or a group of volumes. The same key is assigned to a local volume (or a group of local volumes) and to a remote volume (or a group of remote volumes). One can arbitrarily define groups of volumes. For example, one may define a group of volumes deploying an entire database. --

Please replace the second paragraph on page 6, with the following amended paragraph:

-- The desired remote disk address can be retrieved from the local disk system. As described previously, the local disk system has stored the relationship between the local disk or volume and the remote disk or volume when the administrator established a pair. This enables the remote disk address to be located. By referring to the appropriate entry in the encryption control table corresponding to the address, the remote disk system locates the key for the disk. The local disk system knows its local disk address. By referring the entry corresponding to the address in the encryption control table 200, it finds the correct key for the disk. Steps ~~310-330~~ 300-330 illustrate locating the right key at the remote disk system. A write request from the local disk system to the remote disk system includes the remote disk address. Once the address is located, the data is sent to the remote disk, decrypted, and stored, all as shown by steps 330-340. When

the write at the remote disk is complete, a message 350 is sent to the local disk system, informing it of the completion. --

Please add the following new paragraph at line 10, after the second paragraph, on page 7:

-- Accordingly, the first method of transparent key exchange is summarized as follows:

Step 410 - Store a new key to a memory and send it to the remote disk system.

Step 420 - Store the new key to a memory.

Step 430 - Get the current I/O number of the volume pair.

Step 440 - Choose the appropriate I/O number (the boundary number) to validate the new key and send it to remote disk system.

Step 450 - Wait for the I/Os with the boundary number; I/Os with the boundary number or smaller are decrypted with the current key.

Step 460 - Wait for the I/Os with the boundary number; I/Os with the boundary number or smaller are encrypted with the current key.

Step 470 - Set the new key to Key210; I/Os with the number greater than the boundary number are encrypted with the new key.

Step 480 - Set the new key to Key210; I/Os with the number greater than the boundary number are decrypted with the new key. --

Please add the following new paragraph at line 1, before the first paragraph, on page 8:

-- Accordingly, the second method of implementing key exchange is summarized as follows:

Step 510 - Store a new key to a memory and send it to the remote disk system.

Step 520 - Store the new key to a memory.

Step 530 - Split the pair (Stop copying data to remote disk system).

Step 540 - Split the pair.

Step 550 - Store the new key to Key210 to validate it.

Step 560 - Store the new key to Key210 to validate it.

Step 570 - Re-synchronize the pair (start copying data to the remote disk system).

Step 580 - Re-synchronize the pair. --

Please add the following new paragraph at line 8, before the second paragraph, on page 8:

-- The encryption and decryption techniques of Fig. 6 are summarized as follows:

Step 610 - Store "encryption = NO and decryption = NO" to a memory and send it to the remote disk system.

Step 620 - Store "encryption = NO and decryption = NO" to a memory.

Step 630 - Get the current I/O number of the volume pair.

Step 640 - Choose the appropriate I/O number (the boundary number) to switch encryption and decryption off and send it to remote disk system.

Step 650 - Wait for the I/Os with the boundary number; I/Os with the boundary number or smaller are decrypted with the current key.

Step 660 - Wait for the I/Os with the boundary number; I/Os with the boundary number or smaller are encrypted with the current key.

Step 670 - Store "NO" to Encryption220 and Decryption230; I/Os with the number greater than the boundary number are not encrypted.

Step 680 - Store "NO" to Encryption220 and Decryption230; I/Os w/the number greater than the boundary number are not decrypted. --

Please add the following new paragraph after the above paragraph and before the second paragraph, on page 8:

-- The encryption and decryption techniques of Fig. 7 are summarized as follows:

Step 710 - Store "encryption = NO and decryption = NO" to a memory and send it to the remote disk system.

Step 720 - Store "encryption = NO and decryption = NO" to a memory.

Step 730 - Split the pair (Stop copying data to remote disk system).

Step 740 - Split the pair.

Step 750 - Store "NO" to Encryption220 and Decryption230.

Step 760 - Store "NO" to Encryption220 and Decryption230.

Step 770 - Re-synchronize the pair (start copying data to the remote disk system).

Step 780 - Re-synchronize the pair. --

Please add the following new paragraph after the above paragraph at line 30, after the third paragraph, on page 8:

-- The transparent key exchange technique of Fig. 8 is summarized as follows:

Step 800 - Set all bits of the re-encryption bitmap to 1 (one).

Step 810 - Copy request exists from the local disk system? If yes, go to Step 890. If no, proceed to step 820.

Step 820 - All bits of the re-encryption bitmap are 0 (zero)? If yes, the process ends. If no, the process proceeds to step 830.

Step 830 - Find the next track whose bit is 1 (one).

Step 840 - Read a track from the disk to the cache.

Step 850 - Decrypt the track by the current key.

Step 860 - Encrypt the track by the new key.

Step 870 - Write the track from the cache to the disk.

Step 880 - Set 0 (zero) to the bit of the re-encryption bitmap.

Step 890 -Do steps 840 to 860 for the track of the request and then execute copy request. --